CSE3013 컴퓨터공학 설계 및 실험1 **실험 CPP-1: 예비 보고서**

20180305 중국문화 고혜정

주어진 문제 및 5-1 의 문제 해결에 관한 내용을 이해하고 이 문제를 효율적으로 해결하기 위한 방법을 생각하여 이를 1 쪽 이내로 요약하여 제출하시오.

문제 해결을 위한 간단한 단계별 수행 내용, 자료구조 등을 기술하시오

1. 문제 이해하기

Array, RangeArray 클래스를 구현하고 클래스 선언에 맞추어 멤버 함수를 구현하고 필요한 연산자들을 사용한다.

2. 해결 방법 구상하기

1) Class Array

- int *data (protected) : 배열에 메모리를 동적 할당하여 자료가 저장된다.
- int len (protected) : 배열의 크기를 저장한다.
- 생성자 Array (int size) (public) : 크기가 size인 배열을 생성한다.
- 소멸자 ~Array ( ) (public) : 배열을 메모리에서 제거한다.
- int length( ) (public) : 배열의 크기를 반환한다.
- int & operator[ ] (int i) (public) : i 값이 0 이상 ~ len 미만일 경우 배열의 i번째 위치에 값을 할당한다.
- int operator[ ] (int i) (public) : i 값이 0 이상 ~ len 미만일 경우 배열의 i번째 위치에 값을 반환한다.
- void print() (public) : 배열을 순회하여 모든 원소를 출력한다.

2) Class RangeArray

- int low (protected) : 배열의 시작 인덱스를 지정한다.
- int high (protected) : 배열의 끝 인덱스를 지정한다.
- RangeArray (int low, int high) (public) : 크기가 high – low + 1 인 배열을 생성한다.
- ~RangeArray( ) (public) : 배열을 메모리에서 제거한다.

- int baseValue( ) (public) : 배열의 시작 인덱스를 반환한다.
- int endValue( ) (public) : 배열의 끝 인덱스를 반환한다.
- int & operator[ ] (int i) (public) : i 값이 low 이상 ~ high 이하일 경우 배열의 i번째 위치에 값을 할당한다.
- int operator[ ] (int i) (public) : i 값이 low 이상 ~ high 이하일 경우 배열의 i번째 위치에 값을 반환한다.

3. 추가 학습 내용

1) OOP (Object Oriented Programming) 이란 무엇인가?

'객체 지향 프로그래밍'은 절차와 알고리즘보다는 객체와 자료구조에 기반을 둔 프로그래밍 패러다임이다. 클래스를 정의하고 각 클래스의 특정한 인스턴스로서 객체를 활용하며, 각각의 객체가 일련의 연관된 작업을 수행하도록 한다. 재사용성 최대화, 데이터 의존성 축소, 동적 결합, 캡슐화, 다형성, 상속 등으로 특징 지어진다.

- 객체 : 클래스에서 정의한 것을 토대로 메모리 (실제 저장공간)에 할당된 것이다. 데이터 또는 식별자에 의해 참조되는 공간이며 변수, 자료구조, 함수 또는 메소드가 될 수 있다.
- 클래스 : 객체를 만들 수 있는 설계도다. 객체를 정의하기 위한 속성과 기능들의 집합이다.
- 인스턴스 : 클래스의 설계대로 만들어진 객체들이다.

2) OOP를 쓰는 이유는?

먼저, 객체 지향 프로그래밍은 프로그램을 유연하고 변경이 용이하게 만들기 때문에 개발과 보수가 편리하며 추상 자료구조를 설계하고 활용하기에 적합하다. 그리고 기존 코드를 개조할 때 기존 코드를 활용하여 새로운 클래스를 만들 수 있기 때문에 수정을 최소화할 수 있다. 또한, 라이브러리를 만들어 다른 프로젝트에서 활용하기에 편리하고 직관적인 코드 분석이 가능하다는 장점이 있다.

3) 상속이란 무엇인가?

새로운 클래스가 기존의 클래스의 자료와 연산을 이용할 수 있게 하는 기능을 바로 '상속'이라고 한다. 상속을 통해서 기존의 클래스를 상속 받은 하위 클래스(자식 클래스, 하위 클래스 등등)를 이용해 프로그램의 요구에 맞추어 클래스 수정이 가능하다. 그리고 클래스 간의 종속 관계를 형성하여 객체를 조직화 할 수 있다.